

Databases 2

Concurrency Control

Alessandro Artale

Faculty of Computer Science – Free University of Bolzano

Room: 221

`artale@inf.unibz.it`

`http://www.inf.unibz.it/~artale/`

2003/2004 – First Semester

Problem 1: Cuncurrency Control

Given the following schedule:

S. $r_1(A); r_1(B); r_2(B); w_2(B); r_1(C); r_3(C); w_3(C); r_2(C); w_1(A); w_1(B)$.

Give the precedence graph and check whether the schedule is conflict-serializable.

Justify each precedence relation in the graph, i.e., for each arch in the obtained graph show a conflicting pair of actions which gives rise to such an arch.

Problem 2: Upgrading Locks

Given the following schedule:

S. $r_1(A); r_2(A); r_2(B); w_2(B); r_1(C); r_3(C); w_1(A); w_3(C); r_2(C)$.

Assuming that the scheduler deals with Two-Phase Locking with shared and exclusive locks and allowing upgrading (i.e., a shared lock can be upgraded to an exclusive lock), add shared and exclusive locks and unlocks to **S**. Insert each shared and exclusive lock as delayed as possible, while unlock as soon as possible (always in accordance with the Two-Phase Locking strategy). Give **S** in a table with three columns T_1, T_2, T_3 .